Stat 350: Lab 4

James Watterson

March 12, 2015

A. Interpretation of a Confidence Interval

Below is a table showing the range of a 95% confidence interval for 30 normal distributions (the 1st and 2nd columns) after is a column indicating whether the range contains the value $\mu = 10$ (1 meaning it does and 0 it does not). In total there are 28 ranges that contain the mean, a number that is expected, because at such a high confidence interval one expects almost all ranges to contain the mean.

1	9.77	11.02	1.00
2	9.25	10.48	1.00
3	9.41	10.77	1.00
4	9.42	10.64	1.00
5	9.36	10.37	1.00
6	9.50	10.47	1.00
7	9.48	10.84	1.00
8	9.60	10.77	1.00
9	9.30	10.63	1.00
10	9.46	10.63	1.00
11	9.42	10.78	1.00
12	9.79	10.97	1.00
13	9.59	10.85	1.00
14	8.93	10.21	1.00
15	9.25	10.58	1.00
16	9.48	10.68	1.00
17	9.14	10.47	1.00
18	9.10	10.32	1.00
19	9.25	10.47	1.00
20	10.10	11.33	0.00
21	8.66	9.96	0.00
22	9.57	10.84	1.00
23	9.36	10.70	1.00
24	9.08	10.35	1.00
25	9.33	10.44	1.00
26	9.29	10.60	1.00
27	9.38	10.78	1.00
28	9.24	10.54	1.00
29	8.71	10.03	1.00
30	9.38	10.67	1.00

B. Water Quality Testing

1. Power of Test Given Parameters

a. For $\alpha = 1\%$ and $\mu_{alt} = 6.5$ the power is 0.8128029

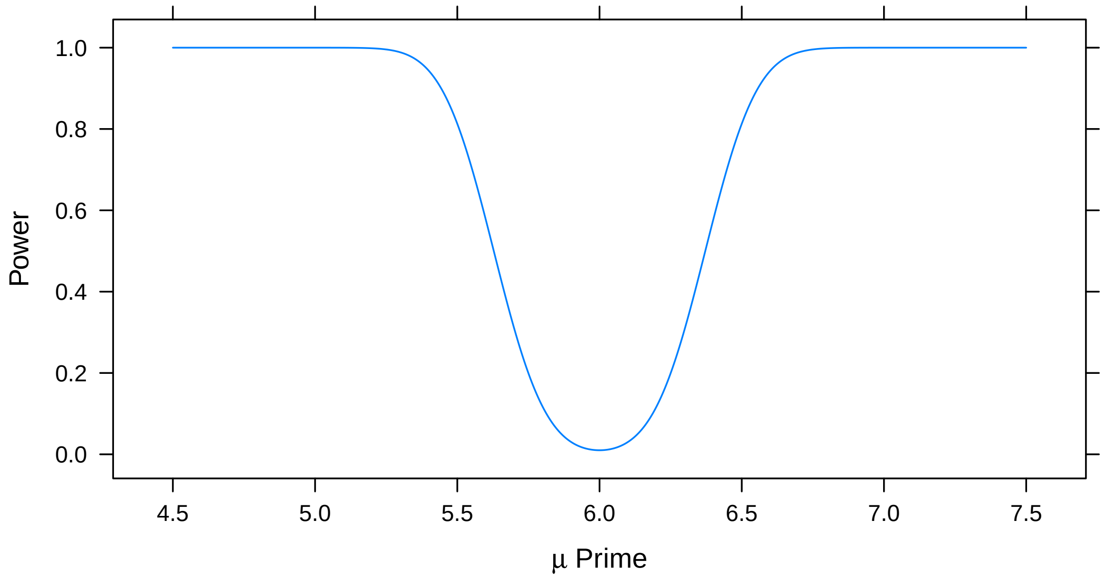
b. For $\alpha = 5\%$ and $\mu_{alt} = 6.5$ the power is 0.9337271

c. For $\alpha = 1\%$ and $\mu_{alt} = 6.75$ the power is 0.9956077

d. For $\alpha = 5\%$ and $\mu_{alt} = 6.5$ with $n = 5$ the power is 0.9710402

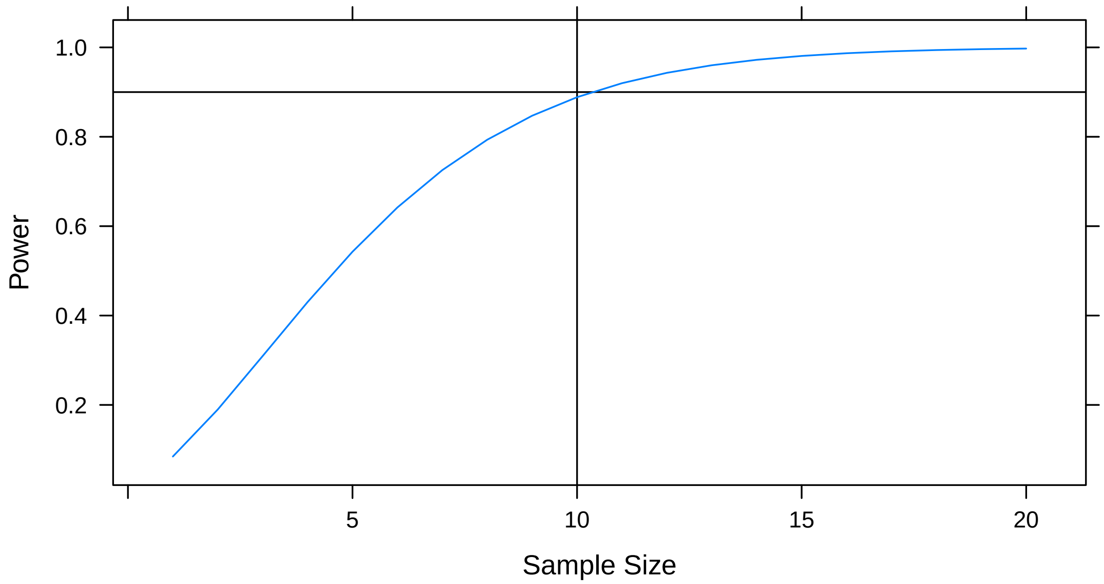
e. Increasing any of the parameters (significance level, alternative μ, and sample size) also increases the power. The degree of this relationship requires more study, but what is clear is that the relationship is positive.

2. Power Curve with Sample Size of 3



3. Desired Sample Size

The sample size required to have a power of atleast 0.90 at $\alpha = 1\%$ with $\mu_{alt} = 6.3$ is 10, see the graph below for proof.



Code

```
library(lattice)
library(xtable)
power_curve <- function(n, a, mu0, std, int_range = mu0 * 0.01 * 2) {
    std_n <- std/sqrt(n)
    z <- qnorm(1 - a/2)
    mu_p <- seq(mu0 - int_range/2, mu0 + int_range/2, mu0 * 1e-04)
    x1 <- mu0 - z * std_n
    x2 <- mu0 + z * std_n
    p_x1 <- pnorm(x1, mu_p, std_n)
    p_x2 <- pnorm(x2, mu_p, std_n, lower.tail = FALSE)
    matrix(c(mu_p, (p_x1 + p_x2)), ncol = 2)
}
specific_power <- function(n, a, mu0, std, mu_a) {
    std_n <- std/sqrt(n)
    z <- qnorm(1 - a/2)
    mu_p <- mu_a
    x1 <- mu0 - z * std_n
    x2 <- mu0 + z * std_n
    p_x1 <- pnorm(x1, mu_p, std_n)
    p_x2 <- pnorm(x2, mu_p, std_n, lower.tail = FALSE)
    return(p_x1 + p_x2)
}
b1a <- specific_power(3, 0.01, 6, 0.25, 6.5)
b1b <- specific_power(3, 0.05, 6, 0.25, 6.5)
b1c <- specific_power(3, 0.01, 6, 0.25, 6.75)
b1d <- specific_power(5, 0.01, 6, 0.25, 6.5)
power_vals <- power_curve(3, 0.01, 6, 0.25, 3)
xyplot(power_vals[, 2] ~ power_vals[, 1], xlab = expression(paste(mu, " Prime")),
    ylab = "Power", size = 1, type = "l")
powers <- sapply(1:20, function(x) specific_power(x, 0.01, 6, 0.25, 6.3))
desired_power <- 0.9
cond_arr <- sapply(powers, function(x) x > desired_power)
# cond_arr starts with false, so the first true is located at size-sum
calc_n_size <- length(cond_arr) - sum(cond_arr)
```